University of Pisa
Business Process Modelling
A.Y 2024-2025

P57 - Cooking School Case

Submitted to:
Prof. Roberto Bruni

Submitted by:
Sara Hoxha (681760)

July 22, 2025

Contents

1 Introduction

This report analyzes and models the business process of a cooking school, described as following:

Consider the scenario of a cooking school that must manage student requests. A student contacts the school and receives a list of available courses. The student selects a course and is put in contact with an instructor. The instructor proposes a date and location for the first lesson, and the student can either accept or propose different date and location options. This iteration continues until an appointment is set. Before each lesson, the student receives a list of ingredients and tools to familiarize themselves with before they are used in the session. During the lesson, the student prepares a series of recipes: the instructor describes the preparation steps, the student begins executing them, and, in case of doubts, asks the instructor for advice. At the end of each session, all dishes are tasted, and the instructor shares her impressions with the student. After making an electronic payment to the school, the student and the instructor can schedule a new appointment (using the protocol described above), or the student can decide to conclude the course. If the course is completed, the client informs the school, and the process is concluded. Modify the processes so that, at the end of the course, the student can choose to start a new learning path.

2 BPMN Modelling

BPMN was chosen to model the scenario as it offers a clear and structured way to represent the collaborative nature of the process and the interactions between different actors. The modeling was carried out using Camunda Modeler, which was selected for its ability to effectively handle complex workflows and visualize inter-participant communication.

The analysis began by identifying the three key participants involved in the process: the *student*, the *cooking school*, and the *instructor*. These actors are represented as separate pools within the BPMN model. Their interactions are modeled through message flows, providing a collaborative diagram that captures the multiple viewpoints of the process. An overview of the core process is shown in Figure 1 below.

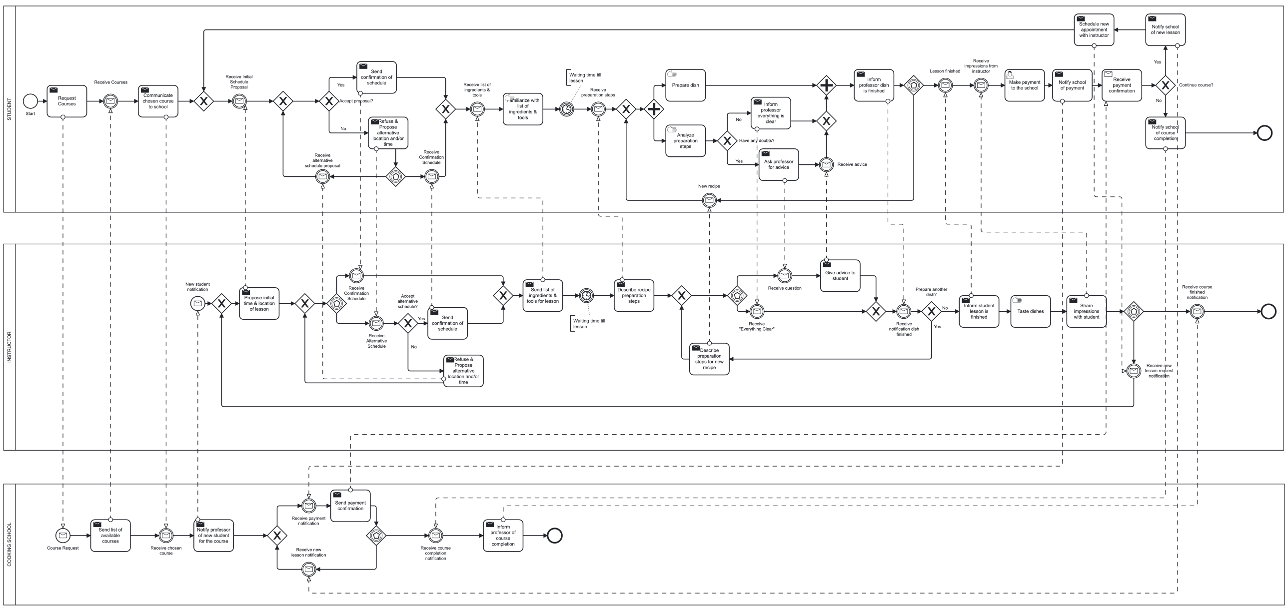


Figure 1: BPMN Base Version

2.1 Student

The process begins in the student pool (see Figure A1), where the student initiates a *Send Message* task to contact the cooking school and request a list of available courses. This is followed by an *intermediate message catching event* representing the reception of the course list from the school. Upon receiving the list, the student selects a preferred course and communicates the choice to the school through another *Send Message* task.

Subsequently, the student receives a proposal from the instructor specifying the time and location of the lesson. At this point, an exclusive (XOR) gateway is used to model a decision: the student may either accept the proposal, in which case the process proceeds, or refuse it and propose an alternative schedule. This alternative is sent via a *Send Message* task, and the student then awaits the instructor's response. The instructor's reply is modeled using an event-based gateway, which branches the process depending on externally received messages, captured by intermediate message catching events. If the instructor accepts the student's alternative proposal, the process continues via an XOR join gateway. If the proposal is rejected, the student receives a counter-proposal and the process reverts to the XOR join, thereby forming an iterative loop for negotiating the lesson schedule until a mutual agreement is reached.

Once the schedule is confirmed, the student receives a list of the required ingredients and tools from the instructor. The student then engages in a *Manual Task* to become familiar with the provided materials. The time gap between receiving the materials and the scheduled lesson is represented by an *intermediate timer event*.

During the lesson, the student receives the recipe's preparation steps through a message catching event. The student then begins to prepare the dish, while concurrently analyzing the steps involved. These two parallel flows are modeled using a parallel (AND) gateway. After analyzing the preparation steps, the student must decide whether any doubts remain, modeled by an XOR gateway. If no doubts arise, the student informs the instructor that everything is clear via a *Send Message* task. If doubts exist, the student requests clarification from the instructor using another *Send Message* task and waits for a response, represented by an intermediate message catching event. The two branches—whether clarification was needed or not—are subsequently joined using an XOR gateway.

Once both preparation and analysis are completed, an AND gateway ensures synchronization before proceeding. The student then notifies the instructor that the dish is complete. Following this, the student awaits a response from the instructor indicating either that the lesson is concluded or that a new dish must be prepared. This branching is managed using an event-based gateway dependent on external messages. If the student is required to prepare a new dish, the process loops back to the XOR join, enabling another iteration of the preparation phase. If the lesson is concluded, the student is notified through a message catching event and then receives feedback from the instructor for the dishes prepared throughout the lesson, again through a message catching event.

The student then completes a *User Task* to make payment to the school and subsequently informs the school of the payment via a *Send Message* task. Afterwards, the student waits to receive a confirmation message that the payment has been received. At this stage, the student is presented with a decision regarding course continuation, modeled by an XOR gateway. If the student chooses not to continue, a message is sent to the school indicating course completion, and the process terminates with an end event. If the student decides to proceed with further lessons, the student informs the school of this and a new appointment is scheduled by contacting the instructor, and the process loops back to the XOR join preceding the initial schedule proposal. This enables an iterative structure that supports continuous learning until the student opts to complete the course.

2.2 Cooking School

The process in the Cooking School pool (see Figure A3) begins upon receipt of a course list request from the student. In response, the school sends a list of available courses to the student via a *Send Message* task and then waits to receive the student's selected course, which is modeled using an intermediate message catching event. Then the school informs the professor of the chosen course of the new student, which is modelled via a *Send Message* task. Following the selection, the school proceeds to wait for the corresponding payment from the student, which is expected after the lesson has been completed. After receiving the payment, the school sends a confirmation message to the student. Subsequently, the school awaits the student's decision on whether to continue the course, a point in the process modeled by an event-based gateway dependent on external input. If the student chooses to continue, the school receives a notification of this decision, and the process returns to the XOR join gateway, thereby initiating an iterative cycle in which the school awaits the next course payment before proceeding. Conversely, if the student informs the school of the decision to terminate the course, the school proceeds by informing the professor of this choice and then the process concludes by transitioning to the end event.

2.3 Instructor

The process in the Instructor pool (see Figure A2) begins upon receipt of a new student notification from the Cooking School, represented by an intermediate message catching event. Subsequently, the instructor sends a proposal to the student, specifying the time and location of the lesson, via a *Send Message* task. At this stage, an *event-based gateway* models the student's externally-driven decision: the student may either accept the proposed schedule, allowing the process to proceed directly, or reject it by submitting a counter-proposal. In the latter case, the counter-proposal is received by the instructor through an *intermediate message catching event*. An *exclusive gateway (XOR)* is then used to model the instructor's decision on whether to accept the student's proposed alternative. If accepted, the process proceeds through an XOR join gateway. If the proposal is rejected, the instructor sends a new counter-proposal via a *Send Message* task, and the process loops back to the XOR join, thereby establishing an iterative negotiation loop until a mutually agreed schedule is reached. Once the schedule is finalized, the instructor sends the student a list of the necessary ingredients and tools via a *Send Message* task. The interval between the dispatch of this material and the beginning of the lesson is represented by an *intermediate timer event*.

During the lesson, the instructor sends the recipe's preparation steps to the student and waits in case clarification is required. This is modeled using an *event-based gateway*, as the next step depends on an external decision by the student. If no questions are raised, the instructor is notified accordingly. If the student has doubts, a clarification request is received through a message catching event, and the instructor responds with appropriate guidance via a *Send Message* task. These two alternative flows, whether or not clarification was needed, are merged using an XOR gateway.

The instructor then receives a message indicating that the student has completed the dish, represented by an intermediate message catching event. An XOR gateway models the instructor's decision on whether the lesson is complete or whether the student must prepare an additional recipe. In the latter case, the process loops back to the XOR join, initiating a new iteration of the recipe preparation phase. If the lesson is deemed complete, the instructor informs the student through a *Send Message* task and proceeds to evaluate the dish, represented by a *Manual Task*. After tasting the dish, the instructor sends feedback to the student via a *Send*

Message task. The process then waits for the student's decision regarding course continuation, modeled using an event-based gateway. If the student has opted not to continue, the instructor is informed by the school of this decision, and the process terminates at an end event. If the student chooses to continue, the instructor receives a message from the student requesting a new lesson, informs the school of course continuation, and the process loops back to the XOR join preceding the initial schedule proposal.

3 Workflow nets

The processes described in Chapter 2 were subsequently converted into workflow nets, a special form of Petri nets. This transformation allows for a more formal and mathematically rigorous representation of the process, enabling thorough analysis and verification of specific behavioral properties. In order to transform from BPMN to workflow nets, the following rules were used:

- Introduce a unique initial place and a unique final place to mark the start and end of the process, respectively.
- Map each task and event in the BPMN diagram to a corresponding transition in the WF-net.
- Represent each sequence flow with a place that connects the corresponding transitions.
- Model event-based gateways by inserting a place followed by multiple alternative transitions, each representing a possible event outcome.
- Encode XOR (exclusive) splits and joins using combinations of places and transitions that ensure mutual exclusivity of the branches.
- Encode AND (parallel) splits and joins using transitions with multiple outgoing arcs (for splits) or multiple incoming arcs (for joins), modeling concurrency.
- To model message flows between actors, places were added to join the transitions that represent sending and receiving activities, enabling synchronization between participants.

After the transformation, we analyze the individual workflow nets for each actor as well as the full workflow net. The analysis was conducted using the WoPeD tool, whereas for the full workflow, Woflan was used due to the high complexity of the net.

3.1 Cooking School

The *Cooking School* net consists of 9 places, 9 transitions and 18 arcs. It is evident from the analysis (see Figure A4a) that the net is a workflow net, since there is only one source place and one sink place, and node reachability is ensured, as each place and transition is positioned on a path leading from the initial place to the final place. The workflow net is properly initialized as it has exactly one token in the initial place. The net can be identified as an S-net, meaning that every transition is has exactly one input place and one output place. The net N^*, the version of N that contains a reset transition, is not a T-net, since there are places whose pre-set or post-set is not a singleton due to the presence of XOR splits/joins and event-based gateways. Moreover, the net qualifies as free-choice, as any two transitions have either disjoint or identical pre-sets. The net is also considered well-structured, as it does not exhibit any PT-handles or

TP-handles. N^* is S-coverable, consisting of one S-component, which means that the net has a positive S-invariant and therefore it is bounded. It is also strongly connected, forming a single strongly connected component. The Coverability Graph (see Figure A7a) contains 9 nodes and 9 edges, and since the net is bounded, it coincides with the Reachability Graph, which is therefore finite. These structural characteristics imply that the net is bounded, live, safe, and sound.

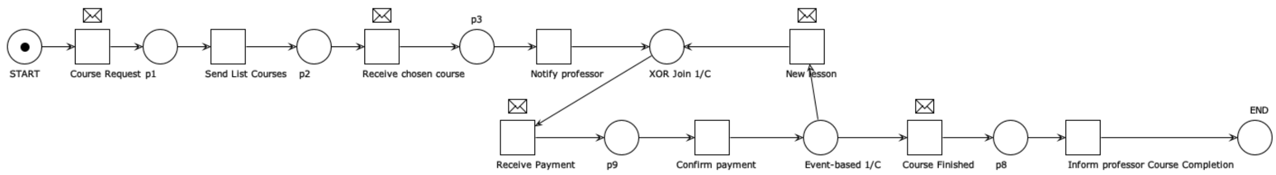


Figure 2: Cooking School Net

3.2 Student

The *Student* net consists of 24 places, 27 transitions and 56 arcs. It is evident from the analysis (see Figure A4b) that the net is a workflow net, since there a single source place and one sink place, and node reachability is ensured, as each place and transition is positioned on a path leading from the initial place to the final place. The workflow net is properly initialized as it has exactly one token in the initial place. The net is not an S-net due to due to the presence of transitions, such as AND-splits and AND-joins, that have multiple input or output places, and thus are not limited to one input and one outplace per transition. N^*, the version of N that contains a reset transition, is not a T-net, since there are places, such as event-based gateways or XOR joins/splits, with multiple input or output transitions, thus violating the T-net condition. The net qualifies as free-choice, as any two transitions have either disjoint or identical pre-sets. The net is also considered well-structured, as it does not exhibit any PT-handles or TP-handles. N^* is S-coverable, consisting of two S-components, which means that the net has a positive S-invariant and therefore it is bounded. It is also strongly connected. The Coverability Graph (see Figure A7b) contains 27 nodes and 56 edges, and since the net is bounded, it coincides with the Reachability Graph, which is therefore finite. These structural characteristics imply that the net is bounded, live, safe, and sound.

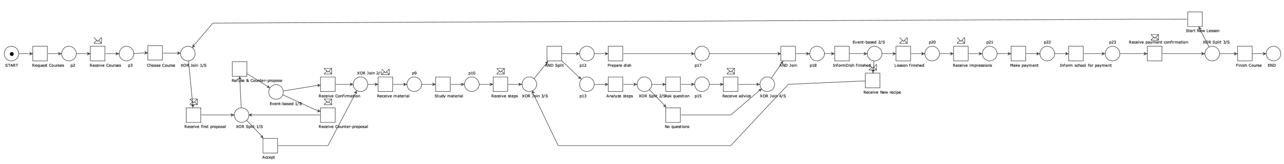


Figure 3: Student Net

3.3 Instructor

The *Instructor* net consists of 14 places, 18 transitions and 36 arcs. It is evident from the analysis (see Figure A4c) that the net is a workflow net, since there is only one source place and one sink place, and node reachability is ensured, as each place and transition is positioned on a path leading from the initial place to the final place. The workflow net is properly initialized as it has exactly one token in the initial place. The net can be identified as an S-net, meaning that every transition has exactly one input place and one output place. The net N^*, the version of N that contains a reset transition, is not a T-net, since there are places, such as event-based gateways or XOR joins/splits, with multiple input or output transitions, thus violating the T-net condition. Moreover, the net qualifies as free-choice, as any two transitions have either disjoint or identical pre-sets. The net is also considered well-structured, as it does not exhibit any PT-handles or TP-handles. The net N^* is S-coverable, consisting of one S-component, which means that the net has a positive S-invariant and therefore it is bounded. It is also strongly connected. The Coverability Graph (see Figure A7c) contains 18 nodes and 36 edges, and since the net is bounded, it coincides with the Reachability Graph, which is therefore finite. These structural characteristics imply that the net is bounded, live, safe, and sound.

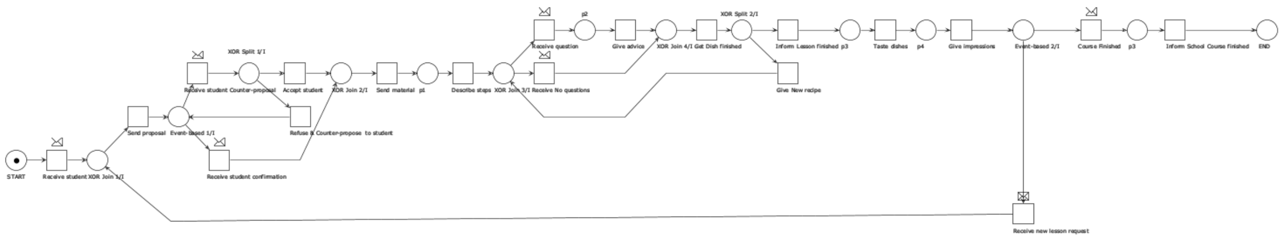


Figure 4: Instructor Net

3.4 Collaboration Worfklow Net

The three workflow nets were merged to construct a comprehensive Petri net that captures the full dynamics of the process (see Figure 5). It is composed of 71 places, 56 transitions, and 162 arcs. The model incorporates all relevant message exchanges between participants and an intermediate place was introduced to connect the distinct transitions across pools. The initial place is unique and belongs to the *Student*, which initiates the process by requesting the list of available courses. The final place is also unique and is reached only after a transition that synchronizes the completion of all other subprocesses, ensuring the workflow concludes only once all roles have fulfilled their respective activities. The net continues to be a sound workflow net, as it is live and bounded, as attested by Woflan analysis (see Figure A6a). The net does not qualify as an S-net, as the Student net, as previously discussed, fails to meet the structural requirement of having exactly one input and one output place per transition. In addition, unlike the other nets, it no longer satisfies the free-choice property, due to the addition of message flows, provide the existence of transitions that don't have disjoint pre-sets. The Coverability Graph, since the net is bounded, it coincides with the Reachability Graph, which is therefore finite. However, the graph is not included in the report due to its large size and poor readability.

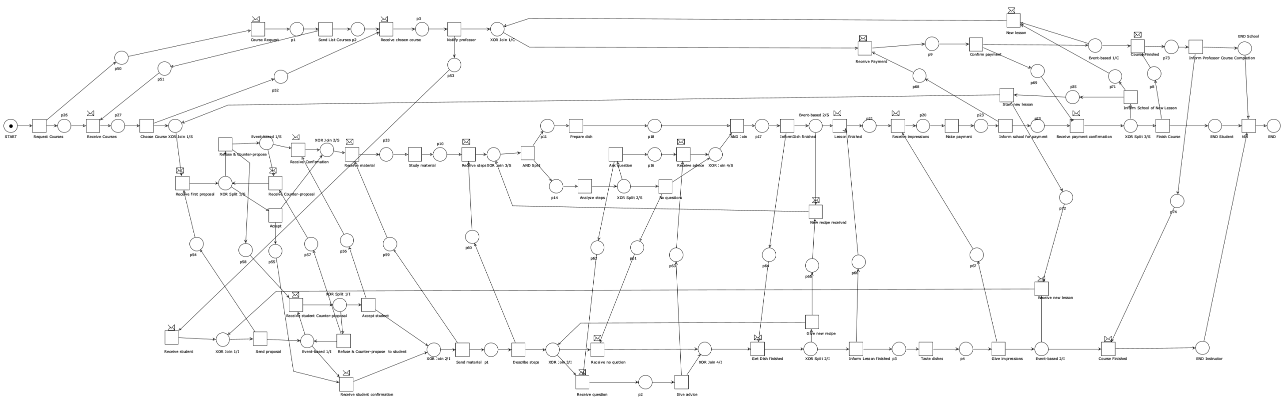


Figure 5: Collaboration Workflow Net

4 Process Variant

The scenario under consideration includes a variant in which the student is given the opportunity to decide whether to begin a new learning path at the end of the course. An BPMN modelling of this variant process is presented in Figure 6. The following changes were made to the base process described in Chapter 2:

In the *Student* pool, following the confirmation that the current course will not continue, the student must decide whether to enroll in a new course. This decision is modeled using an *exclusive gateway (XOR)*. If the student chooses not to begin a new course, they inform the school accordingly via a *Send Message* task, and the process concludes with an *End Event*. Conversely, if the student wishes to start a new course, a message is sent to the Cooking School, and the process loops back to a new XOR gateway that precedes the course request, thus enabling an iterative structure that supports the initiation of successive learning paths until the student chooses to terminate.

In the *Cooking School* pool, the student's decision is captured using an *event-based gateway*, as it depends on an external input. Two flows are modeled: if the student decides not to continue, the school receives this decision via an *intermediate message catching event*, informs the professor of this and the process terminates. If the student opts to begin a new course, school informs the professor that a new professor will be required for the new course, the process returns to the XOR gateway that precedes the sending of available course options, thereby forming a loop that supports continuous course scheduling.

In the *Instructor* pool, a new event-based gateway linked to an *intermediate message catching event*, which waits for a message regarding the new course status from the school in case a new teacher is required, was added. This event is followed by a gateway that redirects the flow back to the beginning of the instructor's process, thus enabling the assignment of a new instructor whenever a new course cycle begins. Otherwise, the process terminates.

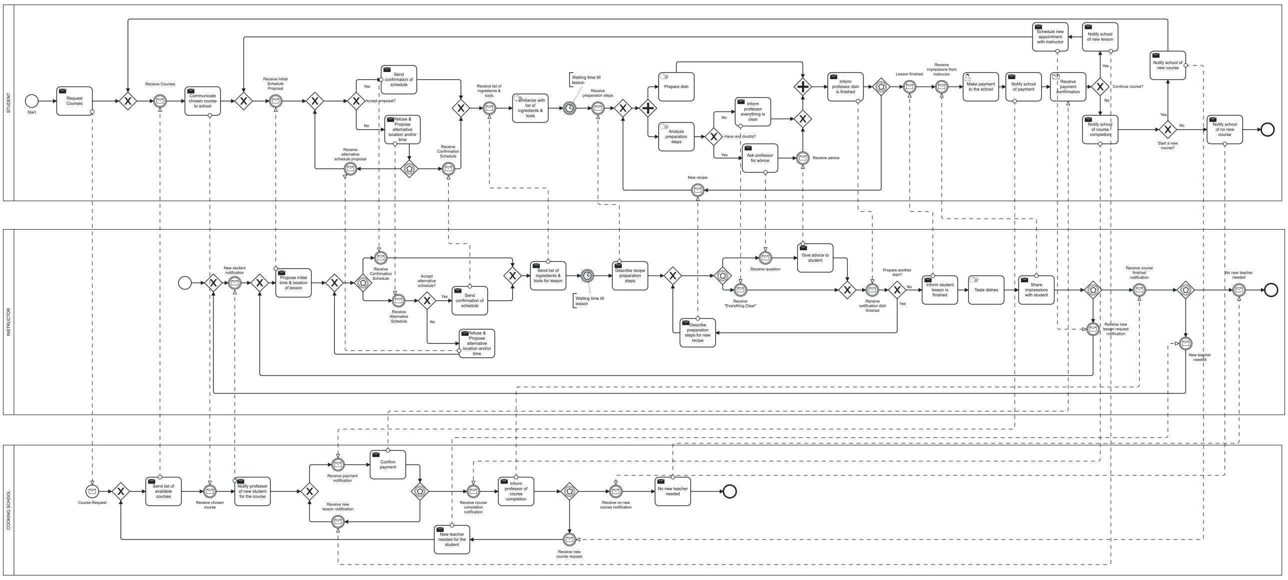


Figure 6: BPMN Variant Version

4.1 Cooking School

The *Cooking School* net in Figure 7, although modified in the variant version, retains all the properties of the previously discussed net. It remains live, bounded, sound, S-coverable (with one component), free-choice, and well-structured as can be seen in the analysis in Figure A5a. As in the original model, it qualifies as an S-net, but not as a T-net. However, the change can be seen in the structural properties of the net which now consists of 12 places, 13 transitions, and 26 arcs. The rechability graph can be seen in Figure A8a, which now has more nodes/arcs due to the expansion of the net.

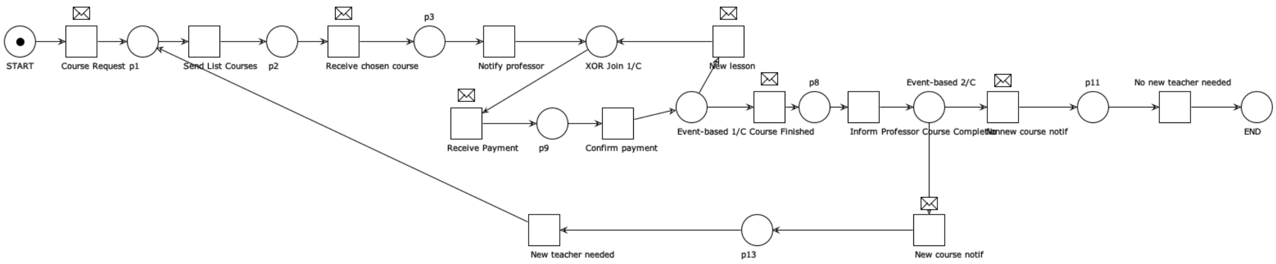


Figure 7: Variant Cooking School Net

4.2 Student

The *Student* net in Figure 8, although modified in the variant version, retains all the properties of the previosly discussed net. It remains live, bounded, sound, S-coverable (with two components), free-choice, and well-structured as can be seen in the analysis in Figure A5b. As in the original model, it does not qualify as a T-net or an S-net. However, the change can be seen in the structural properties of the net which now consists of 25 places, 29 transitions, and 60 arcs. The rechability graph can be seen in Figure A8b, which now has more nodes/arcs due to the expansion of the net.

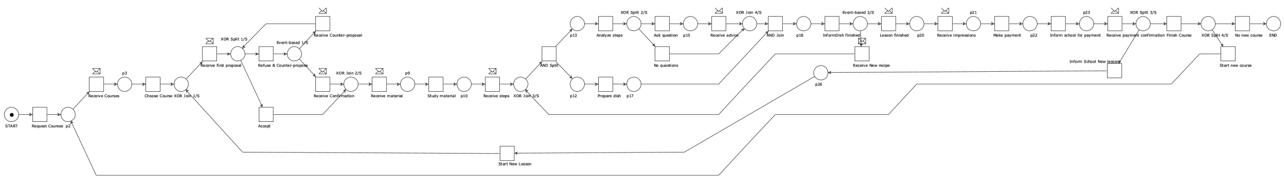


Figure 8: Variant Student Net

4.3 Instructor

The *Instructor* net in Figure 9, although modified in the variant version, retains all the properties of the previously discussed net. It remains live, bounded, sound, S-coverable (with one component), free-choice, and well-structured as can be seen in the analysis in Figure A5b. As in the original model, it qualifies as an S-net, but not as a T-net. However, the change can be seen in the structural properties of the net which now consists of 17 places, 22 transitions, and 44 arcs. The rechability graph can be seen in Figure A8c, which now has more nodes/arcs due to the expansion of the net.

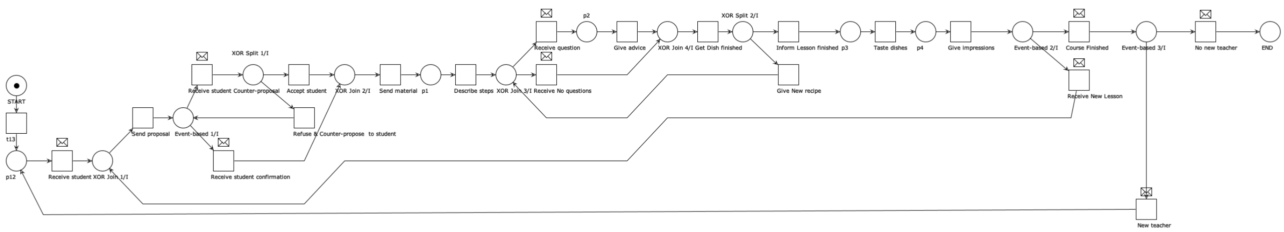


Figure 9: Variant Instructor Net

4.4 Collaboration Worfklow Net

The three workflow nets of the variant model were merged to construct a comprehensive Petri net (see Figure 10). It is composed of 83 places, 66 transitions, and 190 arcs. The model incorporates all relevant message exchanges between participants and an intermediate place was introduced to connect the distinct transitions across pools. Just like in the other collaboration workflow net, the initial place is unique and belongs to the *Student* and the final place is also unique and is reached only after a transition that synchronizes the completion of all other subprocesses. The net continues to be a sound workflow net, as it is live and bounded, as attested by Woflan analysis (see Figure A6b). It continues to not qualify as a S-net or free-choice for the same reasons explained in the previous section. Again, the Coverability Graph, since the net is bounded, it coincides with the Reachability Graph, which is therefore finite. However, the graph is not included in the report due to its large size and poor readability.

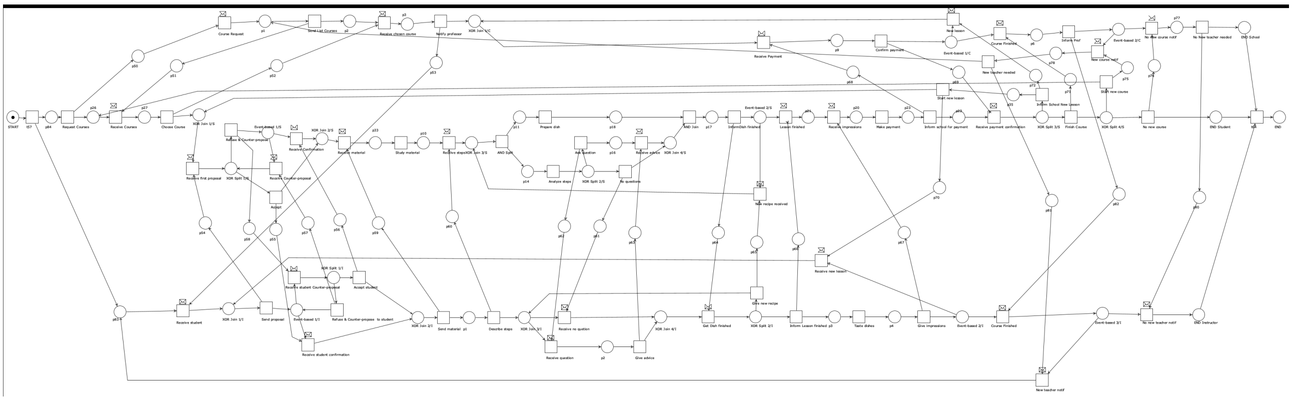


Figure 10: Variant Collaboration Workflow Net

A Appendix A

A.1 BPMN Pools

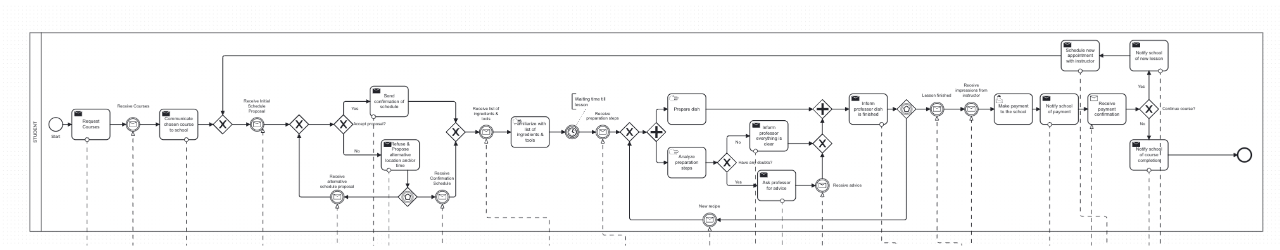


Figure A1: Student Pool

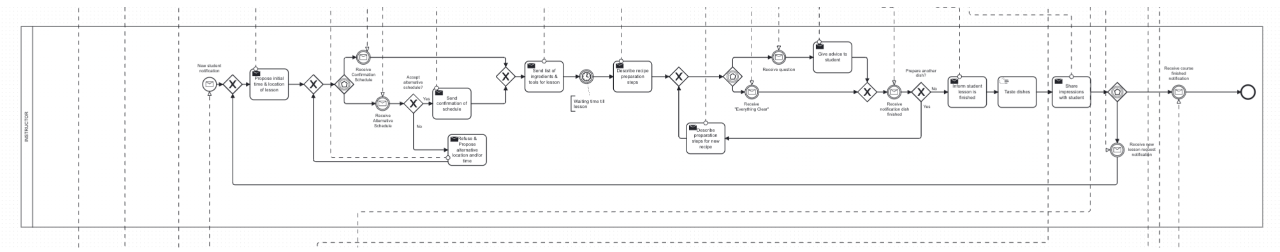


Figure A2: Instructor Pool

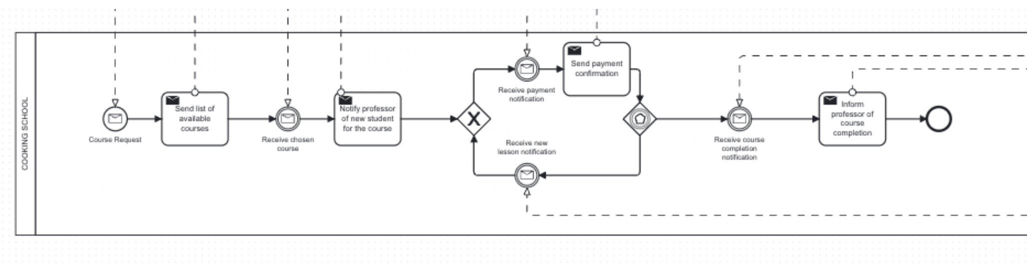
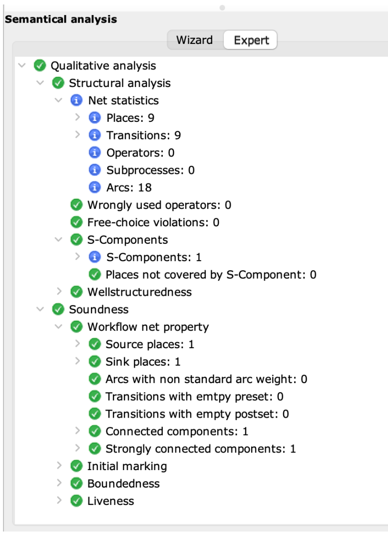


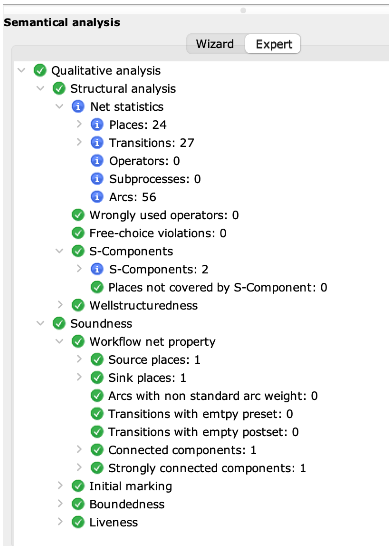
Figure A3: Cooking School Pool

A.2 Workflow Nets

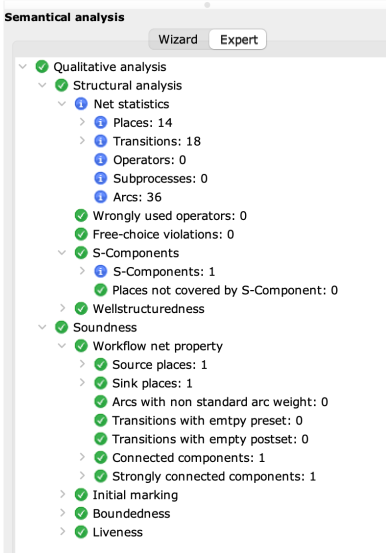
A.2.1 Workflow Nets Analysis



(a) Cooking School Analysis

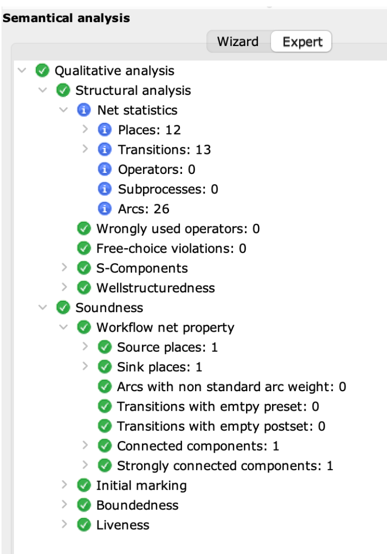


(b) Student Analysis

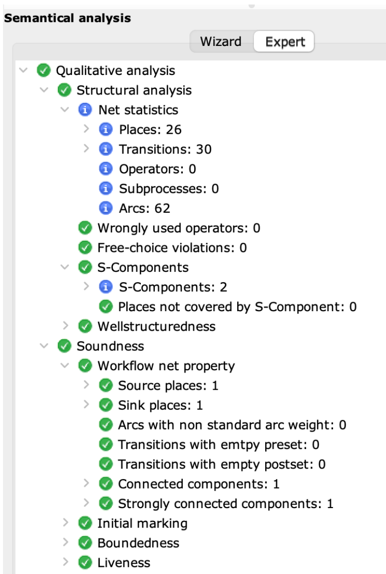


(c) Instructor Analysis

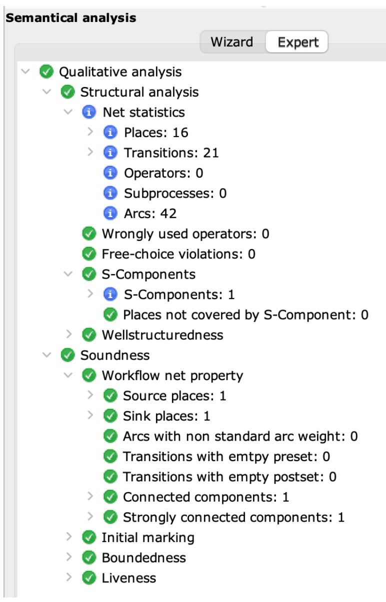
A.2.2 Variant Workflow Nets Analysis



(a) Variant Cooking School Analysis

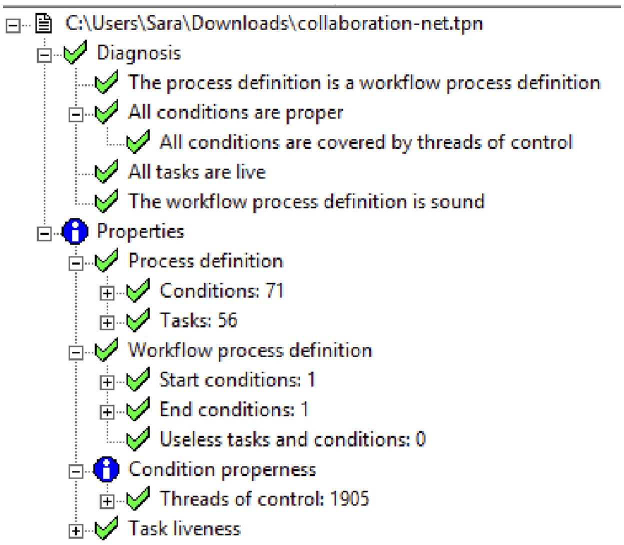


(b) Variant Student Analysis

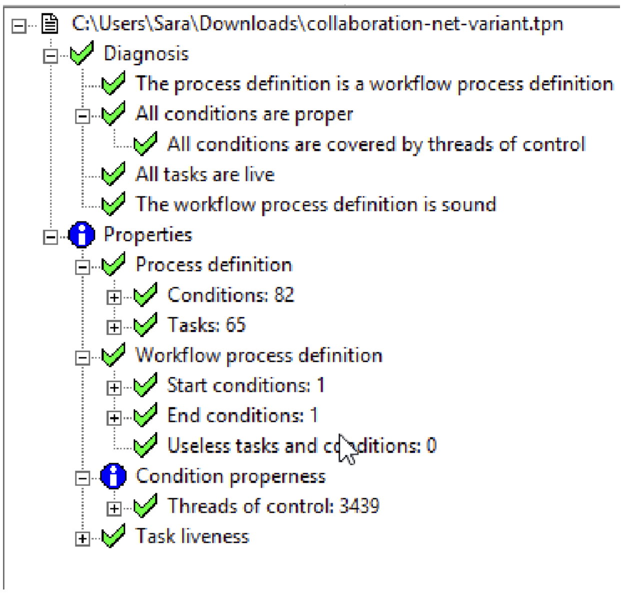


(c) Variant Instructor Analysis

A.2.3 Woflan Collaboration Nets Analysis

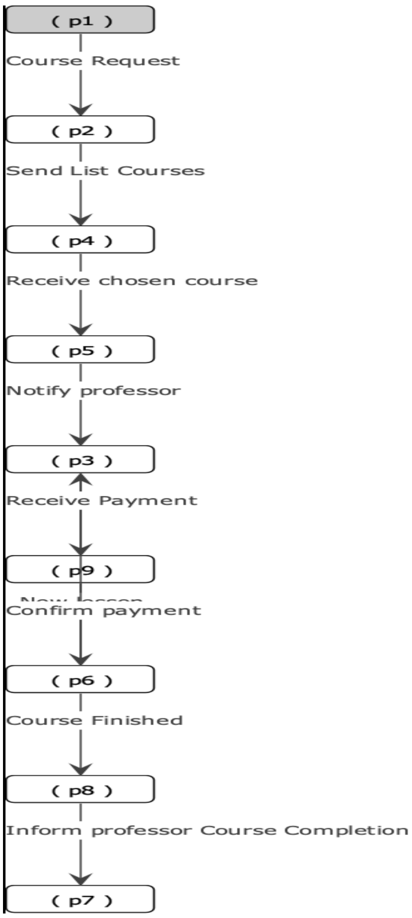


(a) Collaboration Woflan Analysis

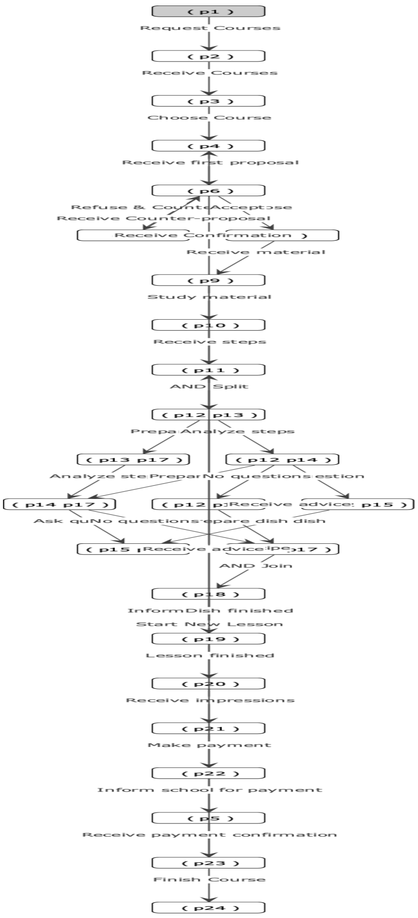


(b) Variant Collaboration Woflan Analysis

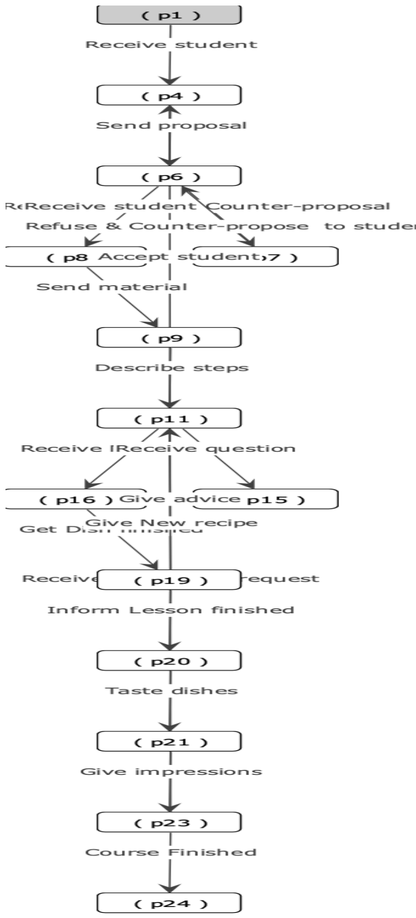
A.2.4 Coverability Graphs



(a) Cooking School

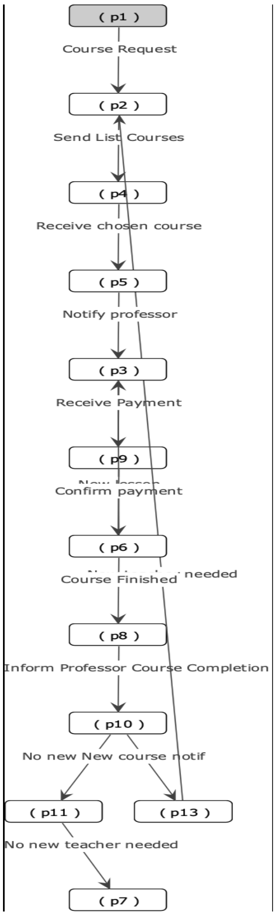


(b) Student

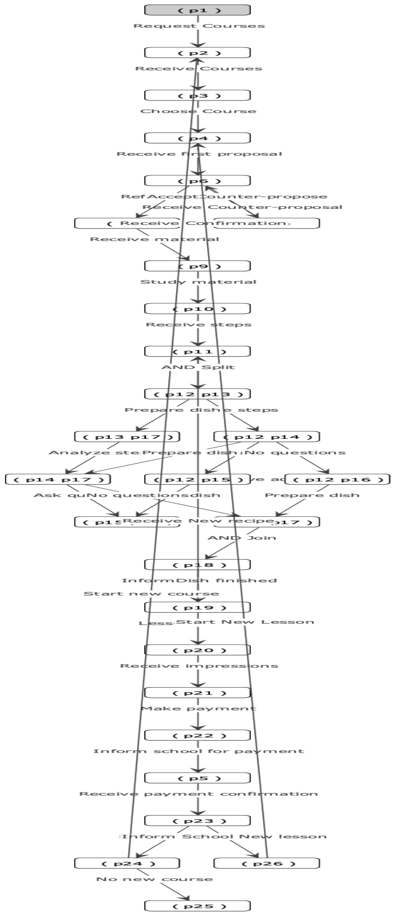


(c) Instructor

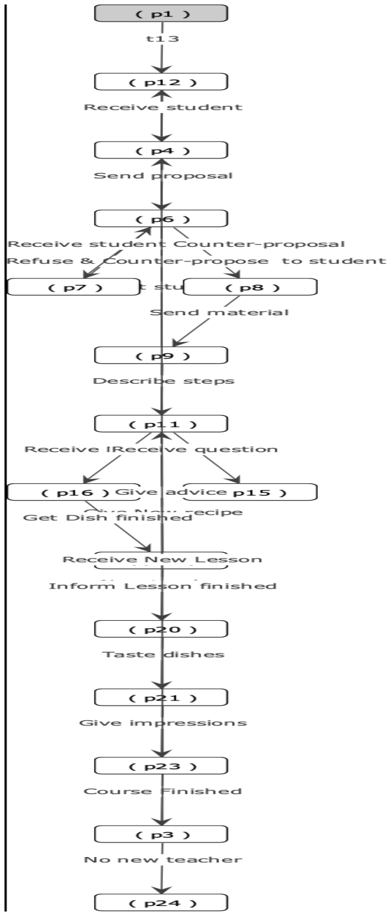
A.2.5 Variant Coverability Graphs



(a) Variant Cooking School Graph



(b) Variant Student Graph



(c) Variant Instructor Graph